**Microsoft PLR 4-2, Exhibit C: Mini Markman Preliminary Claim Construction – Claim Phrases**[1]

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| '193:1 | Claim as a Whole: The recited method is performed within a **VDE**. |
| receiving a digital file including music | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The qualifier "including music" recites a non-functional descriptive material and is not a patentable limitation.<br>3. The recited function requires: obtaining a **VDE secure container encapsulating a digital file, authenticating** the intended recipient in accordance with **VDE controls associated with the secure container**, and accepting the **secure container**. |
| a budget specifying the number of copies which can be made of said digital file | 1. A **budget** identifying the total number of **copies** (whether or not decrypted, long-lived, or accessible) that (since creation of the budget) can be made of the **digital file** by any and all users, devices, and processes. No process, user, or device is able to make another **copy** of the **digital file** once this number of **copies** has been made. |
| controlling the copies made of said digital file | 1. **Controlling uses** of and **accesses** to all **copies** of the **digital file**, by all users, processes, and devices, by executing each of the recited "at least one" **copy control(s)** within **VDE Secure Processing Environment(s)**. Each **control governs (controls)** only one action, which action may or may not differ among the different "at least one" **controls**. All **uses** and **accesses** are prohibited and incapable of occurring except to the extent **allowed** by the "at least one" **copy control(s)**. |
| determining whether said digital file may be copied and stored on a second device based on at least said copy control | 1. Determining whether this particular first device is **allowed** to perform both of the following actions on this particular **digital file**: (1) **copy** it and (2) store it (as opposed to a **copy** of it) on a second device, by executing one or more **VDE control**(s) (including "said" **copy control associated with this digital file**) within **VDE Secure Processing Environment**(s). To the extent that either of these two actions is not determined by this step to be permissible, that action is absolutely prohibited and incapable of occurring, and no user, process or device can perform it on this **digital file**. |

[1] The word "invention" is used not to suggest that anything described in InterTrust's patents in fact was novel or non-obvious or inventive, but rather to identify what was described as the **alleged** invention. Also, features and capabilities are described as they are described in the InterTrust patent application, even though the patent application did not describe an actual working system having any of these capabilities. Also, Microsoft's proposed constructions use many terms from the InterTrust patents that are used inconsistently or otherwise indefinitely in the patents. Those terms are used by Microsoft in their narrowest applicable sense, and without waiving the right to assert the indefiniteness of this claim language. Also, the preliminary constructions assume (without conceding) that the February, 1995, InterTrust patent application was incorporated by reference into the '721, '861, and '683 patents, effectively for claim construction purposes. If the Court concludes otherwise, then the proper constructions will be different in some cases. Bolded terms are preliminarily defined in Exhibits A-C of Microsoft's PLR 4-2 papers.

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| | 2. This claim limitation's recitation of "said" **copy control** is inconsistent with the claim limitation "at least one" **copy control.** |
| if said copy control allows at least a portion of said digital file to be copied and stored on a second device | 1. This "if" condition creates two branches for the recited process, each of which must be performed. Each time the "if" condition is met, all four of the later-recited actions (copying, transferring, storing, playing) must occur. Each time it is not met, each of these four actions must be prohibited and incapable of occurring.<br>2. This "if" condition is met if and only if "said" **copy control allows** any portion of the **digital file** to be **copied** and also **allows** that same **portion** of the **file** (as opposed to the **copy**) to be stored on any second device. This "if" condition is based entirely on "said copy control" and thus is met, as above, even if other **VDE control(s)** prohibit those actions.<br>3. This claim limitation's recitation of "copy control allows at least a portion" is inconsistent with the claim limitation "whether said digital file may be copied … based on at least said copy control." |
| copying at least a portion of said digital file | 1. **Copying** at least some portion of the **digital file** (as opposed to a **copy** thereof), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). This copied "portion" may or may not be (or even include) the portion referred to in the claim limitation "if said copy control allows at least a portion." |
| transferring at least a portion of said digital file to a second device | 1. Transferring to some second device (which may or may not be the "second device" referred to in the claim limitation "if said copy control allows at least a portion of said digital file to be copied and stored on a second device") at least some portion of the **digital file** (as opposed to a **copy** thereof), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). This transferred portion may or may not be (or even include) the portion referred to in the claim limitation "if said copy control allows at least a portion," or the portion referred to in the claim limitation "copying at least a portion." |
| storing said digital file | 1. Storing the entire **digital file** received in the "receiving" step (as opposed to a **copy** of the **file** or an incomplete portion of the **file**).<br>2. This claim limitation's recitation of "storing said digital file" is inconsistent with the claim limitation "transferring at least a portion of said digital file." |
| '193:11 | Claim as a Whole: The recited method is performed within a **VDE.** |
| receiving a digital file | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The recited function requires: obtaining a **VDE secure container encapsulating a digital file, authenticating** the intended recipient in accordance with **VDE controls associated with** the **secure container**, and accepting the **secure container.** |
| determining whether said digital file may be copied | 1. Determining whether said first **control**, by itself, **allows** this particular first device to perform both of the following actions on this particular **digital file**: (1) **copy** it and (2) store it (as opposed to **a copy** of it) on a second device, by executing the first **VDE control** within **VDE Secure Processing** |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| and stored on a second device based on said first control | **Environment**(s). To the extent that either the **copy** or store action is not determined by this step to be permissible, that action is absolutely prohibited and incapable of occurring, and no user, process or device can perform it on this **digital file**. |
| identifying said second device | 1. Identifying a second device sufficiently to distinguish it from all other devices, by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). |
| whether said first control allows transfer of said copied file to said second device | 1. Whether the first **control**, by itself, **allows** the entire **digital file** (which has been copied at least once) (as opposed to the copy) to be moved to the identified second device. If not, that transfer is absolutely prohibited and incapable of occurring and no user, process or device can perform that action on this **file**. |
| said determination based at least in part on the features present at the device | 1. Basing the determination at least in part upon all actual, current features of the device (as opposed to previously determined, reported, or measured features) which might affect the device's ability to prevent unauthorized **access** to and/or **use** of the **digital file**. This determination is done without trusting either the device or any user of the device. A device identifier such as a serial number is not a "feature present at the device." |
| if said first control allows at least a portion of said digital file to be copied and stored on a second device | 1. This "if" condition creates two branches for the recited process, each of which must be performed. Each time the "if" condition is met, all four of the later-recited actions (copying, transferring, storing, rendering) must occur. Each time it is not met, each of these four actions must be disabled and prohibited and incapable of occurring.<br>2. This "if" condition is met if and only if the first **control** allows any portion of the **digital file** to be **copied** and also allows that same **portion** of the **file** (as opposed to the copy) to be on any second device. This "if" condition is based entirely on the first **control** and thus is met, as above, even if other **VDE controls** prohibit those actions.<br>3. This claim limitation's recitation of "said first control allows at least a portion" is inconsistent with the claim limitation "whether said digital file may be copied ... based on said first control." |
| copying at least a portion of said digital file | 1. **Copying** at least some portion of the **digital file** (as opposed to a **copy** thereof), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). The copied portion may or may not be (or even include) the portion referred to in the claim limitation "if said first control allows at least a portion." |
| transferring at least a portion of said digital file to a second device | 1. Transferring to some second device (which may or may not be the "second device" referred to in the claim limitation "if said first control allows at least a portion of said digital file to be copied and stored on a second device") at least some portion of the **digital file** (as opposed to a **copy** thereof), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). The transferred portion may or may not be (or even include) the portion referred to in the claim limitation "if said first control allows at least a portion," or the portion referred to in the claim limitation "copying at least a portion." |
| storing said digital file | 1. Storing the entire **digital file** received in the "receiving" step (as opposed to a **copy** of the **file** or an incomplete portion of the **file**). |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| | 2. This claim limitation's recitation of "storing said digital file" is inconsistent with the claim limitation "transferring at least a portion of said digital file." |
| '193:15 | Claim as a Whole: The recited method is performed within a **VDE**. |
| receiving a digital file | 1. See 193:11. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step). |
| an authentication step comprising: | 1. **Authenticating** the first device and/or user of the first device without relying on trusting either, by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). |
| accessing at least one identifier associated with a first device or with a user of said first device | 1. **Securely accessing** at least one **identifier associated with** a single ("first") device or (as opposed to "and") with a single, current user of that device, by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). One of the "at least one identifier" may be **associated with** a first device while another of the "at least one identifier" may be **associated with** a user of said first device. |
| determining whether said identifier is associated with a device and/or user authorized to store said digital file | 1. For each accessed "at least one **identifier**," determining whether the device with which it is **associated** is one on which the **file** may be stored (by any user) and/or whether the user with which it is **associated** is one who may store the **file** (on any device), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). Each **identifier** may be **associated with** a device "and" a user, or with a device only, or with a user only.<br>2. This claim limitation's recitation of "said identifier" is inconsistent with the claim limitation "at least one identifier." |
| storing said digital file in a first secure memory of said first device, but only if said device and/or user is so authorized, but not proceeding with said storing if said device and/or user is not authorized | 1. This conditional step creates at least two "**authentication**" branches for the recited process, each of which must be performed. Each time the condition is met, the recited "storing" must occur. Each time it is not met, the recited "storing" must not occur.<br>2. If "storing" proceeds, then: storing in a **secure memory** of the first device, the entire **file** received in the "receiving" step, as opposed to a **copy** of the **file** or an incomplete portion of the **file**, by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). If "storing" does not proceed: then the **file** is not stored in the **secure memory** of the first device, and is absolutely prevented from being stored anywhere on the first device.<br>3. This limitation is internally inconsistent on the circumstances under which the storing proceeds or does not proceed. For example, the first ("only if") phrase requires that the storing step proceeds if the device is authorized (and the user is not) while the second ("but not") phrase requires that the storing step not proceed if the device is authorized (and the user is not). |
| storing information associated with said digital file in a secure database stored on said first device, said information including | 1. Storing information in a **secure database**, the entirety of information (including the "at least one **control**") being **associated with** the **digital file** (as opposed to the **file's** contents independent of the **file**), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s).<br>2. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step). |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| at least one control | |
| determining whether said digital file may be copied and stored on a second device based on said at least one control | 1. Determining whether the "at least one control," by itself or themselves, **allow**(s) this particular first device to perform both of the following actions on this particular **digital file**: (1) **copy** it and (2) store it (as opposed to **a copy** of it) on a second device, by executing "said at least one control," by executing the "at least one" **VDE control** within **VDE Secure Processing Environment**(s). To the extent that either the **copy** or store action is not determined by this step to be permissible, that action is absolutely prohibited and incapable of occurring, and no user, process or device can perform it on this **digital file**.<br>2. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step). |
| if said at least one control allows at least a portion of said digital file to be copied and stored on a second device, | 1. This "if" condition creates two branches for each of the two "authentication branches" of the recited process (and thus four branches in all), each of which must be performed. Each time it is met, all four of the later-recited actions (copying, transferring, storing, rendering) must occur. Each time it is not met, each of these four actions must be prohibited and incapable of occurring.<br>2. This "if" condition is met if and only if the at least one **control allows** any portion of the **digital file** to be **copied** and also allows that same portion of the **file** (as opposed to the **copy**) to be stored on any second device. This "if" condition is based entirely on the at least one **control** and thus is met, as above, even if other VDE **controls** prohibit those actions.<br>3. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step).<br>4. This claim limitation's recitation of "at least one control allows at least a portion of said digital file" is inconsistent with the claim limitation "whether said digital file may be copied ... based on said at least one control." |
| copying at least a portion of said digital file | 1. **Copying** at least some portion of the **digital file** (as opposed to a **copy** thereof), which portion may or may not be (or even include) the portion referred to in the claim limitation "if said at least one control allows at least a portion," by executing **VDE control**(s) within **VDE Secure Processing Environment**(s).<br>2. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step). |
| transferring at least a portion of said digital file to a second device | 1. Transferring to some second device (which may or may not be the "second device" referred to in the claim limitation "if said at least one control allows at least a portion of said digital file to be copied and stored on a second device") at least some portion of the **digital file** (not a copy thereof), by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). The transferred portion may or may not be (or even include) the portion referred to in the claim limitation "if said at least one control allows at least a portion," or the portion referred to the claim limitation "copying at least a portion."<br>2. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step). |
| storing said digital file | 1. Storing the entire **digital file** received in the "receiving" step (as opposed to a **copy** of the **file** or an incomplete portion of the **file**). |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| | 2. This step must proceed in both "**authentication** branches" of the process (i.e., regardless of the outcome of the "**authentication**" step).<br>3. This claim limitation's recitation of "storing said digital file" is inconsistent with the claim limitation "transferring at least a portion of said digital file." |
| '193:19 | Claim as a Whole: The recited method is performed within a **VDE**. |
| receiving a digital file at a first device | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The recited function requires: obtaining a **VDE secure container encapsulating** a **digital file, authenticating** the first device in accordance with **VDE controls associated with** the **secure container**, and accepting the **secure container**. |
| establishing communication between said first device and a clearinghouse located at a location remote from said first device | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("establishing communication") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step Part of the recited function is performed by the Remote Procedure Call Manager 732 software of Rights Operating System 602 that controls I/O controller 660 and Communications Controller 666.<br>2. The recited function is: creating and using a previously non-existent communications channel which is necessary and sufficient for exchanging information between the first device and a **clearinghouse**. |
| using said authorization information to gain access to or make at least one use of said first digital file | 1. A user, process or device uses all of said **authorization information** in connection with executing **VDE control**(s) within **VDE Secure Processing Environment**(s) to gain **access** to or (as opposed to "and") make at least one **use** of the **file** received in the "receiving" step. Without using such authorization information, no **access** to or **use** of the file is **allowed**. |
| including using said key to decrypt at least a portion of said first digital file | 1. The "at least one use of said digital file" must encompass decrypting at least a portion of the **digital file** using the **key**. |
| receiving a first control from said clearinghouse at said first device | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The recited function requires: obtaining a **VDE secure container encapsulating** a **first control, authenticating** the first device in accordance with **VDE controls associated with** the **secure container**, and accepting the **secure container**. |
| storing said first digital file | 1. Storing in a **memory** of the first device, the entire **digital file** (as opposed to any incomplete portion thereof) received in the "receiving" step, by |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| in a memory of said first device | executing **VDE control(s)** within **VDE Secure Processing Environment(s)**. |
| using said first control to determine whether said first digital file may be copied and stored on a second device | 1. Determining whether the first **control**, by itself, allows this particular first device to perform both of the following actions on this particular **digital file**: (1) **copy** it and (2) store it (as opposed to a **copy** of it) on a second device, by executing the first **VDE control** within **VDE Secure Processing Environment(s)**. To the extent that either the **copy** or store action is not determined by this step to be permissible, that action is absolutely prohibited and incapable of occurring, and no user, process or device can perform it on this **digital file**. |
| if said first control allows at least a portion of said first digital file to be copied and stored on a second device | 1. This "if" condition creates two branches for the recited process, each of which must be performed. Each time the "if" condition is met, all four of the later-recited actions (copying, transferring, storing, rendering) must occur. Each time it is not met, each of these four actions must be prohibited and incapable of occurring.<br>2. This "if" condition is met if and only if the first **control** allows any portion of the first **digital file** to be **copied** and also allows that same **portion** of the **file** (as opposed to the copy) to be stored on any second device. This "if" condition is based entirely on the first control and thus is met, as above, even if other VDE **controls** prohibit those actions.<br>3. This claim limitation's recitation of "first control allows at least a portion of said first digital file" is inconsistent with the claim limitation "whether said first digital file may be copied ... on a second device." |
| copying at least a portion of said first digital file | 1. **Copying** at least some portion of the **digital file** (as opposed to a **copy** thereof), which portion may or may not be (or even include) the portion referred to in the claim limitation "if said first control allows at least a portion," by executing **VDE control(s)** within **VDE Secure Processing Environment(s)**. |
| transferring at least a portion of said first digital file to a second device including a memory and an audio and/or video output | 1. Transferring to some second device (which may or may not be the "second device" referred to in the claim limitation "if said first control allows at least a portion of said digital file to be copied and stored on a second device") at least some portion of the **digital file** (not a copy thereof), by executing **VDE control(s)** within **VDE Secure Processing Environment(s)**. The transferred portion may or may not be (or even include) the portion referred to in the claim limitation "if said first control allows at least a portion," or the portion referred to the above limitation "copying at least a portion." |
| storing said first digital file portion | 1. Storing the "at least a portion" which was transferred to the second device, of the **digital file** received in the "receiving" step (as opposed to a **copy** of the **file**). |
| '683:2 | Claim as a Whole: The "system" is a **VDE**. |
| user controls | 1. [This shall be construed as a disputed claim term.] |
| the first secure container having been received from a | 1. The "first secure container" must **identify** the single apparatus from which it was received, and that apparatus must be different from the first apparatus. Alternatively, if the Court does not construe this claim language as requiring the "first secure container" to identify the single apparatus |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| second apparatus | from which it was received: This claim language has no patentable weight. It recites a step taken in the creation of the recited system, not a structural or functional characteristic of the system. One studying a particular system (as opposed to the process by which it was created) to compare it to the claimed system, could not distinguish a **secure container** received from another apparatus from, e.g., a **secure container** created on the first apparatus, and thus could not determine whether this step was satisfied.<br><br>2. Receiving the **secure container** includes **authenticating** the intended recipient in accordance with **VDE controls associated with** the **secure container**. The first **secure container** may be received as bar codes in a fax transmission, or filled ovals on a form delivered through physical mail. |
| an aspect of access to or use of | 1. Any one (as opposed to more than one) aspect of any **access** to or (as opposed to "and") **use** by any and all processes, users, and devices. |
| the first secure container rule having been received from a third apparatus different from said second apparatus | 1. The "first secure container rule" must have been received **encapsulated** within a **VDE secure container**, and the intended recipient must have been **authenticated** in accordance with **VDE controls associated with** the **secure container**, and the "first secure container rule" must have been accepted by the first apparatus. The "first secure container rule" must identify the single apparatus from which it was received, and that apparatus must be different from the first apparatus.<br><br>2. Alternatively, if the Court does not construe this claim language as requiring the "first secure container" to identify the single apparatus from which it was received: This claim language has no patentable weight. It recites a step taken in the creation of the recited system, not a structural or functional characteristic of the system. One studying a particular system (as opposed to the process by which it was created) to compare it to the claimed system, could not distinguish a **secure container rule** received from another apparatus from, e.g., a **secure container rule** created on the first apparatus, and thus could not determine whether this step was satisfied. |
| hardware or software used for receiving and opening secure containers | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function (e.g., "**opening**") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br><br>2. The recited function requires: the same single logical piece of either hardware or software (as opposed to both) must be capable of both receiving and **opening secure containers**, this "receiving" including **authenticating** the intended recipient in accordance with **VDE controls associated with** the **secure container**, and this "opening" performed by executing **VDE control(s)** within **VDE Secure Processing Environment(s)**. |
| said secure containers each including the capacity to contain a governed item, a secure container rule being associated with each of said secure containers | 1. Each **secure container** which the "hardware or software used for receiving and opening secure containers" is capable of receiving and **opening** must have the capacity to **contain** a **governed item**, and must have **associated with** it (as opposed to any particular **governed item**) a **secure container rule**. |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| protected processing environment at least in part protecting information contained in said protected processing environment from tampering by a user of said first apparatus | 1. A single **VDE Secure Processing Environment**, in addition to and not within the first apparatus, actively **preventing** (not merely being capable of **preventing**, and not merely resisting) any "user" of the first apparatus from **tampering** with any and all information **encapsulated** by the **Secure Processing Environment** (as opposed to **tampering** with the **Secure Processing Environment** itself). Other components may or may not provide part of this **protecting** function.<br><br>2. The **protecting** function is provided by use of the disclosed "**component assembly**" (VDE **controls**), "**secure container**," "**protected processing environment**," "object registration," and other mechanisms of the purported "**VDE**" "invention" for allegedly individually ensuring the "access control" "handcuffs" between specific "**controls**," specific "objects" (and their content at an arbitrary granular level), and specific "users." |
| hardware or software used for applying said first secure container rule and a second secure container rule in combination to at least in part govern at least one aspect of access to or use of a governed item contained in a secure container | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function ("applying ... in combination") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br><br>2. The recited function requires: a single logical piece of either hardware or software (as opposed to both) to apply the two separate **rules** in combination by assembling and executing a single **control**, and to **govern** any one or more aspects of any **access** or **use** by any process or user or device, of a **governed item contained** in a **secure container** (which may or may not be any "secure container" recited earlier). Other components may or may not provide part of the **governing** function. This "hardware or software" performs its functions by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). |
| hardware or software used for transmission of secure containers to other apparatuses or for the receipt of secure containers from other apparatuses. | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites an undefined mechanism ("hardware or software") for performing a function (e.g., "transmission") without reciting particular structure that performs that function. The specification does not clearly link any particular structure to this recited function. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br><br>2. The recited function requires: a single logical piece of either hardware or software (as opposed to both) is capable of both transmission and receipt of **secure containers**, this receipt including **authenticating** the intended recipient in accordance with **VDE controls associated with** the **secure container**. This "hardware or software" is separate from and in addition to the first apparatus, the recited **protected processing environment**, and the recited "hardware or software used for receiving and opening secure containers." The transmission and receipt of the **secure containers** may be via bar codes in a fax transmission, or filled ovals on a form delivered through physical mail. This "hardware or software" performs its functions by executing **VDE control**(s) within **VDE Secure Processing Environment**(s). |
| **'721:1** | Claim as a Whole: The recited method is performed within a **VDE**. |
| digitally signing a first load module with a first digital signature designating the | 1. **Digitally signing** a particular ("first") **load module** by using a first **digital signature** as the signature **key**, which signing indicates to any and all devices in the first device class that the signor authorized this **load module** for use by that device. No **VDE** device can perform any execution of any **load module** without such authorization. The method ensures that the **load module** cannot execute in a particular device class and ensures that no |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| first load module for use by a first device class | device in that device class has the key(s) necessary to verify the **digital signature**. |
| digitally signing a second load module with a second digital signature different from the first digital signature, the second digital signature designating the second load module for use by a second device class having at least one of tamper resistance and security level different from the at least one of tamper resistance and security level of the first device class | 1. **Digitally signing** a different ("second") **load module** by using a different ("second") **digital signature** as the signature key, which signing indicates to any and all devices in the second device class that the signor authorized this **load module** for use by that device. No **VDE** device can perform any execution of any **load module** without such authorization. The method ensures that the **load module** cannot execute in a particular device class and ensures that no device in that device class has the key(s) necessary to verify the digital signature.<br>2. All devices in the first device class have the same persistent (not just occasional) and identified level of **tamper resistance** and/or same persistent and identified **level of security**. All devices in the second device class have the same persistent and identified level of **tamper resistance** and/or same persistent and identified **level of security**. The identified level of **tamper resistance** and/or identified **level of security** for the first device class, is greater or less than the identified **level of tamper resistance** and/or identified **level of security** for the second device class. |
| distributing the first load module for use by at least one device in the first device class | 1. The first **load module, digitally signed** as indicated above, is transmitted to at least one device in the first device class. |
| distributing the second load module for use by at least one device in the second device class | 1. The second **load module, digitally signed** as indicated above, is transmitted to at least one deviee in the second device class. |
| **'721:34** | Claim as a Whole: The "protected processing environment" is part of and within **VDE**. |
| arrangement within the first tamper resistant barrier | 1. The **arrangement** is located and executed wholly within the first **tamper resistant barrier**. |
| prevents the first secure execution space from executing the same executable accessed by a second secure execution | 1. "A second secure execution space having a second tamper resistant barrier with a second **security level** different from the first security level": a second **secure execution space** (different from the first **secure execution space**) is part of the **protected processing environment**, and has a **tamper resistant barrier** (different from the first **tamper resistant barrier**) which has a persistent (not just occasional) **security level** greater or less than the first persistent **security level**. |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| space having a second tamper resistant barrier with a second security level different from the first security level | 2. "The same executable accessed by": the same **executable** (as opposed to, e.g., two **copies** of the same **executable**) is simultaneously accessed by both the first **secure execution space** and the second **secure execution space**.<br>3. "Prevents the first secure execution space from executing": the **arrangement prevents** the first **secure execution space**, otherwise capable of executing the **executable**, from executing any part of the **executable** (e.g., on behalf of any user, process, or device). |
| **'861:58** | Claim as a Whole: The recited method is performed within a **VDE**. |
| creating a first secure container | 1. This preamble language is a claim limitation.<br>2. Completely forming (as opposed to defining) a **secure container** within a **VDE Secure Processing Environment(s)**. |
| including or addressing . . . organization information . . . desired organization of a content section. . . and metadata information at least in part specifying at least one step required or desired in creation of said first secure container | 1. The same single **descriptive data structure** must either **contain** within its confines or **address** both **organization information** and **metadata information**.<br>2. Both the "desired" organization of the content section and also the "desired" step, occur after the **descriptive data structure** is accessed, not before.<br>3. The **metadata information** specifies a procedure, as opposed to a result or a **data item**. |
| at least in part determine specific information required to be included in said first secure container contents | 1. The **metadata information** is used to determine the specific value, not merely the kind, of at least some of the information that must be placed inside the **secure container**.<br>2. The use of the **metadata information** actively requires the **secure container** creation steps to add this specific information to the first **secure container**, as opposed to the specific information being within the **secure container** for some other reason. |
| rule designed to control at least one aspect of access to or use of at least a portion of said first secure container contents | 1. A **rule** designed for these particular **secure container** contents, which is used (by **VDE control(s)** executing in **VDE Secure Processing Environment(s)**) to limit **access** to or **use** of at least a portion of the contents of the first **secure container** (by all users, processes, and devices). Without compliance with this **rule**, no process, user, or device is able to take the **controlled** aspect of the **controlled access** or **use** action. |
| **'891:1** | Claim as a Whole: The recited method is performed within a **VDE**. |
| resource processed in a secure operating | 1. This preamble language is a claim limitation.<br>2. A component part of a first appliance's **secure operating environment** which is processed within that **secure operating environment**'s special- |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| environment at a first appliance | purpose **Secure Processing Unit**. A **Secure Processing Unit** is a special-purpose unit isolated from the rest of the world in which a hardware **tamper-resistant barrier encapsulates** a processor and internal **secure memory**. The **barrier** prevents all **unauthorized** interference, removal, observation, and **use** of the information and processes within it. The processor cryptographically verifies the **integrity** of all code loaded from the **secure memory** prior to execution, executes only the code that the processor has **authenticated** for its use, and is otherwise **secure**. |
| securely receiving a first entity's control at said first appliance | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("securely receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The recited function requires: A first appliance obtaining a **VDE secure container encapsulating** a **control** created, selected, or modified by a first entity, as part of a communication encrypted on the communications level, **authenticating** the first appliance in accordance with **VDE controls associated with** the **secure container**, and accepting the **secure container**. |
| securely receiving a second entity's control at said first appliance | 1. This claim language falls within 35 U.S.C. § 112, ¶ 6. It recites a step or result ("securely receiving") without reciting an action that achieves that result. The specification does not clearly link any particular action to this recited step. Part of the recited function is performed by Communications Controller 666, I/O Controller 600, SPE 503/SPU 500 (particularly "SPU Encryption/Decryption Engine 522" and NVRAM 534b).<br>2. The recited function requires: A first appliance obtaining a **VDE secure container encapsulating** a **control** created, selected, or modified by a second entity, as part of a communication encrypted on the communications level, **authenticating** the first appliance in accordance with **VDE controls associated with** the **secure container**, and accepting the **secure container**. |
| securely processing a data item at said first appliance, using at least one resource | 1. Performing an operation, inside the special-purpose **Secure Processing Unit** of the first appliance, on a **data item** inside the **Secure Processing Unit**. The operation cannot be observed from outside the SPU and is performed only after the **integrity** of the program code for performing such operation is cryptographically verified. A **Secure Processing Unit** is a special-purpose unit isolated from the rest of the world in which a hardware **tamper-resistant barrier encapsulates** a processor and internal **secure memory**. The **barrier** prevents all **unauthorized** interference, removal, observation, and **use** of the information and processes within it. The processor cryptographically verifies the **integrity** of all code loaded from the **secure memory** prior to execution, executes only the code that the processor has **authenticated** for its use, and is otherwise **secure**. |
| securely applying, at said first appliance through use of said at least one resource said first entity's control and said second entity's control to govern use of said data item | 1. Processing the resource (component part of a first appliance's **secure operating environment**) within the **secure operating environment**'s special-purpose **Secure Processing Unit** to execute the first **control** and second **control** in combination within the SPU. This execution of these **controls** **governs** all use of the **data item** by all users, processes, and devices. The processing of the **resource** and execution of the **controls** cannot be observed from outside the SPU and is performed only after the **integrity** of the **resource** and **controls** is cryptographically verified. A **Secure Processing Unit** is a special-purpose unit isolated from the rest of the world in which a hardware **tamper-resistant barrier encapsulates** a processor and internal **secure memory**. The **barrier** prevents all **unauthorized** interference, removal, observation, and **use** of the information and processes within it. The processor cryptographically verifies the **integrity** of all code loaded from the **secure memory** prior to execution, executes only the code that the processor has **authenticated** for its use, and is otherwise **secure**. |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| '900:155 | Claim as a Whole: The "virtual distribution environment" is **VDE**. |
| first host processing environment comprising | 1. A **host processing environment** that encompasses the recited computer hardware (central processing unit, main **memory**, and mass storage) and certain **VDE Protected Processing Environment** software loaded in that main **memory** and executing in that central processing unit, but does not encompass software, such as the recited **tamper resistant software**, which is stored in mass storage and not executing. |
| said mass storage storing tamper resistant software | 1. The **tamper resistant software** is physically stored within, as opposed to being merely **addressed** by, the mass storage. |
| designed to be loaded into said main memory and executed by said central processing unit | 1. The **tamper resistant software** is capable of being loaded into only said main memory and is capable of being executed only by said central processing unit. |
| said tamper resistant software comprising: . . . one or more storage locations storing said information | 1. The **tamper resistant software** within said mass storage includes one or more storage locations within it. These storage locations are designated to store, and must store, information **derived** by the **machine check programming**, and must not store any other information. |
| derives information from one or more aspects of said host processing environment, | 1. **Deriving** from the **host processing environment** hardware one or more values that uniquely and persistently **identify** the **host processing environment** and distinguish it from other **host processing environments**.<br>2. The "one or more aspects of said host processing environment" are distinguishing components or parts of the **host processing environment** itself, as opposed to, e.g., data or programs stored within the mass storage or main memory, or processes executing within the **host processing environment**. |
| one or more storage locations storing said information | 1. One or more logical storage locations within the **tamper resistant software** storing only information **derived** by the **machine check programming**. |
| information previously stored in said one or more storage locations | 1. Any information once stored in said "one of more storage locations storing said information," but not stored therein when the recited comparison occurs. |
| generates an indication based on the result of said comparison | 1. Producing an indication based solely on the result of the "compares" step. There are only two possible indications: the comparison found an exact match, or it did not. The "indication" need not be displayed to a user. |
| programming which takes one or more actions based | 1. **Executable programming** code that is a part of the **tamper resistant software**, when executed, and not a part of the **host processing environment**. Whenever the recited indication is generated, no matter what it indicates, this code (executing on the CPU for which it was designed |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| on the state of said indication | and loaded in the **memory** for which it was designed) must take an action, or more than one action. The particular action(s) taken must be based solely on the state of that indication. |
| at least temporarily halting further processing | 1. The action(s) taken by this programming must encompass **halting** or temporarily **halting** all further processing of the **host processing environment** and any processes running within it. |
| '912:8 | Claim as a Whole: The recited method is performed within a **VDE**. |
| identifying at least one aspect of an execution space required for use and/or execution of the load module | 1. Defining fully, without reference to any other information, at least one of the persistent features (aspects) of an **execution space** that are **required** for any use, and/or for any execution, of the **load module**. **An execution space** without all of those **required** aspects is incapable of making any such use (e.g., copying, displaying, printing) and/or execution of the **load module**. |
| said execution space identifier provides the capability for distinguishing between execution spaces providing a higher level of security and execution spaces providing a lower level of security | 1. The **execution space identifier**, by itself, provides the **load module** with the capability of determining the persistent **level of security** of any **execution space** in which it is loaded, and of distinguishing between any two **execution spaces** based on their respective, determined persistent (not just occasional) "**levels of security**." This capability extends to at least two **execution spaces** providing a higher **level of security** and at least two **execution spaces** providing a lower **level of security**. |
| checking said record for validity prior to performing said executing step | Before executing any **executable programming** encompassed within any element which is directly or indirectly identified by any information **contained** within the first **record**, evaluating, within a **VDE Secure Processing Environment**, the values and formats of all data fields within the first **record** and confirming that they have legitimate values and formats. |
| '912:35 | Claim as a Whole: The recited method is performed within a **VDE**. |
| received in a secure container | 1. The first **processing environment** obtained a **VDE secure container encapsulating** the record inside, and **authenticated** the intended recipient in accordance with **VDE controls associated with** the **secure container**, and accepted the **secure container**. |
| said component assembly allowing access to or use of specified information | 1. The **component assembly** identifies specific information over which it (by itself and with no other information), executing in a **VDE Secure Processing Environment, allows access** or **use** (as opposed to **access** "and" **use**). Unless **allowed** by the **component assembly**, no user, process, or device is able to **access** or **use** the specified information. The **component assembly** is **associated with** and dedicated to this particular specified information. |
| said first component | 1. The first **record** by itself **contains** sufficient information to unambiguously **identify** the assembled **component assembly**, including all of its |

| Claim Phrase | Microsoft's Preliminary Construction |
|---|---|
| assembly specified by said first record | elements.<br>2. This limitation is inconsistent with the recitation "first record containing identification information directly or indirectly identifying one or more elements of first component assembly." |